Run-time power and performance scaling with CPU-FPGA hybrids

Dr Jose Nunez-Yanez, Mr Arash Beldachi

Department of Electronic and Electrical Engineering

University of Bristol, BS8 1UB

Bristol, UK

Phone : + 44 0117 3315128

j.l.nunez-yanez@bristol.ac.uk

***Abstract*— This paper investigates how a wide dynamic range of performance and power levels can be obtained in commercially available state-of-the-art hybrid FPGAs that include embedded processors. Adaptive voltage and frequency scaling obtained with embedded in-situ detectors is employed to scale performance and power in the FPGA fabric under processor control. The results show that it is possible to obtain energy savings higher than 60% or alternatively double performance at nominal energy. The available voltage and frequency margins create a large number of performance and energy states with scaling possible at run-time with low overheads.**

Keywords— FPGA, energy efficient design, adaptive voltage scaling, energy propotional computing

I. INTRODUCTION

Energy and power efficiency in Field Programmable Gate Arrays (FPGAs) has been estimated to be up to one order of magnitude worse than in ASICs [1] and this limits their applicability in energy constraint applications. In ASICS lowering the supply voltage reduces both dynamic and static power at the cost of increased circuit delay and a similar approach can be used with FPGAs as well [2]. Voltage scaling is often combined with frequency scaling in order to compensate for the variation of circuit delay. Essentially, voltage and frequency scaling attempts to exploit performance margins so that tasks complete just in time obtaining power and energy savings. An example of this is Dynamic Voltage and Frequency Scaling (DVFS) which is a technique that uses a number of pre-evaluated voltage and frequency operational points to scale power, energy and performance. With DVFS, margins for worst case process and environmental variability are still maintained since it operates in an open-loop configuration. However, worst case variability is rarely the case. Previous work [2] has validated an approach called *Elongate* based on in-situ detectors that uses the logic available in the FPGA slices of Xilinx 65 nm Virtex-5 devices to adjust voltage and frequency.

In comparison to that work the main contributions of this paper are:

1. We investigate the compatibility of *Elongate* with newer 28 nm Zynq chips that use a high performance, low power (HPL) process and show its benefits in a realistic video processing application.
2. The presence of the PMBUS (power manager bus) and mixed mode clock managers are used to create a novel adaptive power scaling system in standard off-the-shelf boards capable of generating hundreds of clock frequencies and voltage configurations on-the-fly.
3. The presence of different voltage domains and hardwired processors in Zynq devices is leveraged to create a Linux software daemon that automatically manages different performance and energy points.

The rest of the paper is structured as follows. Section 2 describes related work. Section 3 presents the power adaptive system architecture while section 4 considers the implementation overheads of this architecture. Section 5 explores the performance and power margins available in Zynq devices. Finally, section 6 presents the final conclusions and future work.

II. RELATED WORK

In order to identify ways of reducing the power consumption in FPGAs, some research has focused on developing new FPGA architectures implementing multi-threshold voltage techniques, multi-Vdd techniques and power gating techniques [3-7]. Other strategies have proposed modifying the map and place&route algorithms to provide power aware implementations [8-10]. This related work is targeted towards FPGA manufacturers and tool designers to adopt in new platforms and design environments. On the other hand, a user level approach is proposed in [11]. A dynamic voltage scaling strategy for commercial FPGAs that aims to minimise power consumption for a giving task is presented in their work. In this methodology, the voltage of the FPGA is controlled by a power supply that can vary the internal voltage of the FPGA. For a given task, the lowest supply voltage of operation is

experimentally derived and at run-time, voltage is adjusted to operate at this critical point. A logic delay measurement circuit is used with an external computer as a feedback control input to adjust the internal voltage of the FPGA (VCCINT) at intervals of 200ms. With this approach, the authors demonstrate power savings from 4% to 54% from the VCCINT supply. The experiments are performed on the Xilinx Virtex 300E-8 device fabricated on a 180nm process technology. The logic delay measurement circuit (LDCM) is an essential part of the system because it is used to measure the device and environmental variation of the critical path of the functionality implemented in the FPGA and it is therefore used to characterise the effects of voltage scaling and provide feedback to the control system. This work is mainly presented as a proof of concept of the power saving capabilities of dynamic voltage scaling on readily available commercial FPGAs and therefore does not focus on efficient implementation strategies to deliver energy and overheads minimisation. A comparable approach also based in delay lines is demonstrated, by the authors in [12]. A dynamic voltage scaling strategy is proposed to minimise energy consumption of an FPGA based processing element, by adjusting first the voltage, then searching for a suitable frequency at which to operate. Again, in this approach, first the critical path of the task under test is identified, then a logic delay measurement circuit is used to track the critical point of operation as voltage and frequency are scaled. Significant savings in power and energy are measured as voltage is scaled from its nominal value of 1.0V down to its limit of 0.6V. Beyond this point, the system fails. Xilinx has also investigated the possibility of using lower voltage levels to save power in their latest family implementing a type of static voltage scaling in [13]. The voltage identification bit available in Virtex-7 allows some devices to operate at 0.9 V instead of the nominal 1 V maintaining nominal performance. During testing, devices that can maintain nominal performance at 0.9 V are programmed with the voltage identification bit set to 1. A board capable of using this feature can read the voltage identification bit and if active can lower the supply to 0.9 V reducing power by around 30%. This is a static configuration that maintains the original level of performance and takes place during boot time in contrast with the dynamic approach investigated in this paper. In-situ detectors located at the end of the critical paths remove the need for delay lines. This technology has been demonstrated in custom processor designs such as those based around ARM Razor [14]. Razor allows timing errors to occur in the main circuit which are detected and corrected re-executing failed instructions. The latest incarnation of Razor uses an optimized flip-flop structure able to detect late transitions that could lead to errors in the flip-flops located in the critical paths. The voltage supply is lower from a nominal voltage of 1.2V (0.13µm CMOS) for a processor design based on the Alpha microarchitecture observing approximately 33% reduction in energy dissipation with a constant error rate of 0.04%. The Razor technology requires changes in the microarchitecture of the processor and it cannot be easily applied to other non-processor based designs. It also uses utilizes a specialized flip-flop. Our work in [2] presents the application of in-situ detectors to commercial FPGAs that deploy arbitrary user designs. The presented approach removes the need of delay lines as done previously in [12] increasing the system robustness and efficiency. Additionally, it only uses the technology primitives already available in the FPGA and it does not require chip fabrication or redesign.

III. Power Adaptive System Architecture

The power adaptive controller is formed by two main IP blocks that correspond to the dynamic voltage scaler (DVS) and the dynamic frequency scaler (DFS) as shown in Fig. 1. These two blocks can be instantiated independently and each one has its own AXI slave interface. This has certain advantages since it means that the technology can be used in different modes depending on the available features on the target board and device. The current prototype targets the ZC702 that implements the power manager bus (PMBUS) with access to all the power rails available for reading and writing. The presence of the PMBUS is required for the DVS unit to work. The DFS unit uses the MMCM (Mixed Mode Clock Managers) blocks to obtain different frequencies at run-time and it does not require other board level components. The following sections describe the features of the DVS and DFS units.

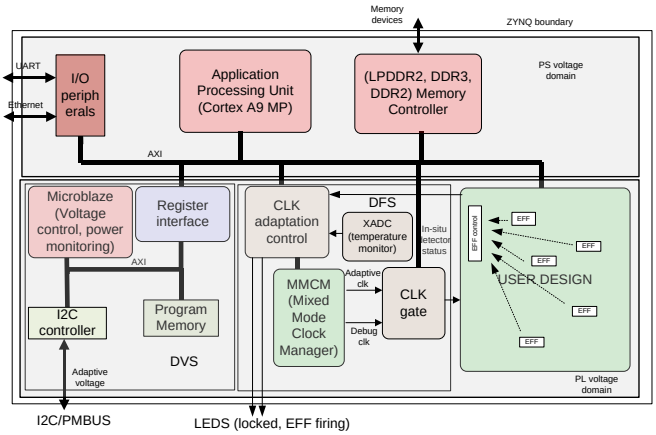

Figure 1. Power adaptive system architecture

A. Dynamic Voltage Scaling unit

As it can be seen in Fig. 1, the DVS unit has three main components which are a MicroBlaze processor (MB), a register file implemented using a Dual-Port RAM (DPRAM) and an IIC IP core. These components are connected to a local AXI bus. The DVS unit has full configuration and monitoring capabilities of the power rails connected to the power manager BUS. The DPRAM is used to receive the commands from the Cortex A9 processors. The commands control and record power, voltage values etc. The MB is responsible for the execution of the commands, communicating with the PMBUS via the IIC IP Core and writing the results to the DPRAM. In the ZC702 board the IIC IP Core is connected to the IIC Bus and accesses the PMBUS through a voltage shifter and an IIC 1-to-8 switch. The initialization code must set the 1-to-8 switch to the PMBUS channel before communication with the voltage regulators is possible. The initialization, configuration and monitoring code is written in C and compiled into a .elf

file using the standard MB compiler. The elf is made part of the bitstream as a firmware and it is automatically stored in the program memory when the device is configured. The functionality of the DVS core is controlled with commands which are issued by Cortex A9 processor. A command has 32 bits and contains six parameters as it can be seen in Fig. 2. Action 1 and 0 are used to activate the core and signal task completion. The rest of the values indicate the type of operation (read/write), the target voltage regulator and the measurement type.

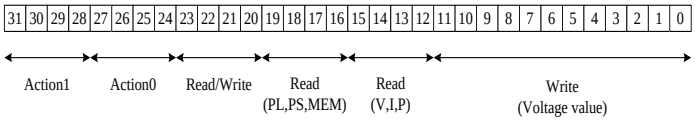

Figure 2. command parameters.

B. Dynamic Frequency Scaling unit

The DFS unit receives the status information of the in-situ detectors embedded in the user design and uses that information to locate the maximum frequency that a particular voltage level can support automatically. A frequency generation ROM memory forms part of the DFS unit. This ROM contains values for the Mixed Mode Clock Managers (MMCM) used to generate the clock for the user logic. The outputs obtained from this memory are written by the state machines part of the DFS unit using the dynamic reconfiguration port available in the MMCM blocks and new frequencies are generated at run-time. Once the MMCM is locked the clock is driven into the user logic. Once the frequency reaches a value that causes timing violations these are reported by the detectors part of Elongate and the state machine stops increasing the frequency until a different voltage is configured in the system. The DFS unit can also instantiate the system monitor IP block available in the FPGA device to monitor internal temperatures. This is advisable so that it is possible to react if internal core temperatures are excessive. Table 2 shows the complexity of the blocks part of the DVS and DFS units.

Table 2- Reference design resources voltage controller/ power monitor (ZC7020 board)

Resource	FF	Utilization	LUT	Utilization
Microblaze processor	972	0.9%	631	1.2%
I2C Controller	343	0.3%	468	0.9%
Clock generation	462	0.4%	683	1.2%
Total	1,777	1.6%	1,782	3.3%
Available	106,400		53,200	

C. Robustness analysis

The power adaptive architecture is designed to search for an optimal frequency for a given voltage value. In the test system the valid range of voltages extends from 0.7 V to 1 V. Frequencies are internally generated using the available MMCM (Mixed Mode Clock Manager) and its capability to reconfigure at run-time. The MMCM dynamic reconfiguration port enables the generation of changes in the clock frequency, phase and duty cycle on the fly. In this work only the clock frequency is varied. There are a number of registers in the MMCM that must be set correctly to control how frequencies are generated and a state machine is required to set the different registers correctly. The important registers in this work control the global clock divider that affects all the clock outputs in the MMCM (range 1 to 128), the individual clock divider for each of the clock outputs (range 1 to 128) and the clock multiplier that changes the voltage control oscillator (VCO) frequency in the MMCM (range 1 to 64).

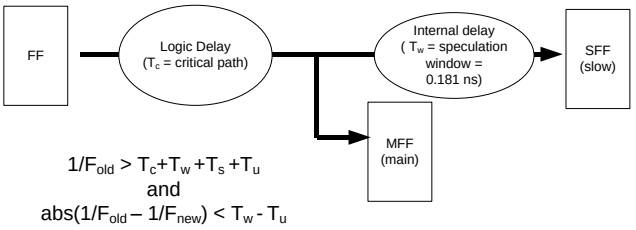

Figure 3. Timing requirements

A problem exists if the instantaneous frequency change (in one single step) is such that both the slow flip-flop and the main flip-flop present in Elongate fail timing and the signal does not land inside the speculation window shown in Fig.3. If this is the case then the system will stop working. Fig.3 shows the timing relations that must hold for the circuit to work. The first equation is the general timing equation and establishes that the clock period has to be large enough to accommodate the logic delay of the main circuit (T_C), the speculation window (T_W), the clock skew (T_S) and the clock uncertainty (T_U). The second equation is specific to *Elongate* and establishes that the change in the clock period between two successive frequencies has to be smaller than $T_W - T_U$ since the clock uncertainty could potentially reduce the speculation window size. Tw is determined by the internal delays in the FPGA slice and calculated using the timing analysis tools to a value of 0.181 ns in the considered technology. Tu is also obtained from the post place&route timing report with a value of 0.035 ns. The *Elongate* tool uses these values as input and calculates the clock frequency generation granularity required in the MMCM to obtain a safe circuit with the additional constraint of maintaining the VCO (Voltage Controlled Oscillator) part of the MMCM within the range allowed by the manufacturer. The possible valid frequencies range from a minimum frequency of 22 MHz to a maximum frequency of 400 MHz. In total 448 different frequencies can be generated and the corresponding register values are stored in a read-only memory using device BRAMS. The CLK generation logic reads these values from the BRAM and writes them to do MMCM in the correct sequence at run-time.

IV. ELONGATE OVERHEADS

For these experiments a user application has been selected in the form of a configurable motion estimation processor suitable for high-definition video coding [15]. The motion estimation core netlist is initially processed to insert the in-situ detectors and implemented together with the rest of the system in the Zynq device. Table 3 shows the additional utilization of the detector logic at around ~1%.

Table 3- Reference design resources original and elongate system (ZC7020 board/ ZYNQ 7020)

Resource	FF	Utilization	LUT	Utilization
Original	16,146	15%	20,914	39%
Elongate	16,380	16%	21,136	40%
Available	106,400		53,200	

Figure 4. Monitoring tool screen capture.

V. POWER AND ENERGY SCALING ANALYSIS

The ARM processor executes a software daemon that reads status information and writes commands to the DFS and DVS units. For these experiments the daemon monitors temperature, frequency, CPU power, FPGA power and detector state. This information is then sent through a USB-UART connection to an external monitoring tool used as a user interface. Fig. 4 shows a capture of the tool at with different frequencies and voltages being generated and detector activity.

The FPGA core voltage is configured with commands written by the daemon to the DVS unit and then the DFS unit is configured by the daemon to search for the highest frequency possible for the given voltage. This point is the most energy efficient point for the given voltage. The DFS unit automatically detects this point and proceeds to inform the daemon. The daemon then restarts the process with a different voltage effectively sweeping the range of valid voltages. Notice that the user application runs in parallel activating the motion estimation processor continuously. This emulates how a real application such as a video codec will make use of a motion estimation accelerator implemented in hardware. The detectors embedded in the user application fire before timing violations affect the motion estimation data paths and control circuits.

A. Power Scaling

Fig.5 shows the valid range of clock frequencies and voltages found by the daemon as it sweeps from nominal voltage of 1.0 V to a low voltage of 0.67 V. The figure shows that there is linear relation between frequency and voltage and, importantly, the detectors fire at a frequency of 255 Mhz at nominal voltage which is much higher than the worst case frequency reported by the tools after timing analysis of 129 MHz. This indicates the existence of performance and power margins that could be exploited depending on workload by this AVS technique.

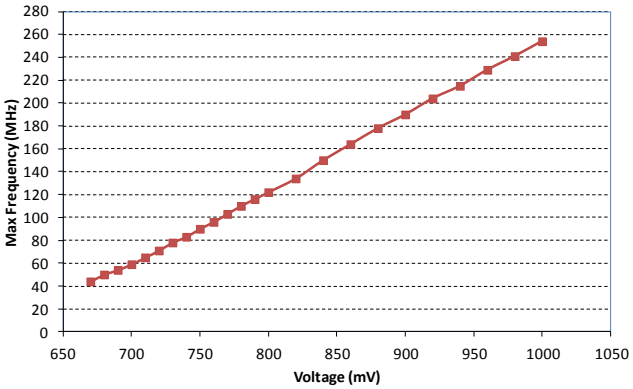

Figure 5. Voltage and frequency.

Equation (1) shows the different power terms in a CMOS device. The first term shows the dynamic power while the second term represents static power dissipation and depends on V_{dd} and I_{leak}. The I_{leak} leakage current has two main components representing the sub-threshold leakage and the gate leakage. For technology generation of 65 nm and below gate leakage can become dominant and is itself heavily dependent on V_{dd}. Fig. 6 shows the effects of scaling voltage in static power in the Zynq fabric. We observe an exponential relation between voltage and static power which confirms that the FPGA fabric can significantly reduce its static power with voltage scaling.

$$Power = \alpha * C * V^2 * f + I_{leak} * V_{dd} \quad (1)$$

In Fig. 7 the motion estimation processor is active and continuously receives activation commands from the processor side. The software daemon reduces the supply voltage via the PMBUS and the maximum frequency supported at each voltage level is auto-detected by the system.

The obtained values define an optimal power profile and this is shown Fig. 7. The nominal power line is based on a fixed nominal voltage of 1.0 V and moves between the max frequency as reported by the tools of 129 MHz to a low frequency of around 40 MHz.

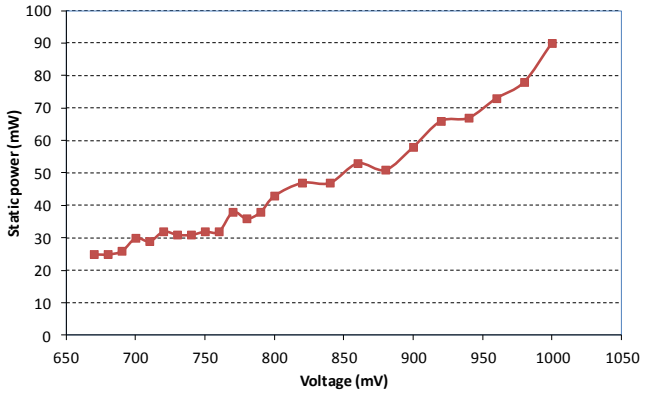

Figure 6. Voltage and frequency.

The figure shows that for a given frequency value the optimal power line is approximately halved. This confirms that power savings between 40% and 60% are possible maintaining the performance. Fig. 8 compares static and dynamic power ratios for the motion estimation core. Static power is higher than dynamic power for the configurations with lower voltage while for the higher voltage configurations static power reduces to approximately 35% of the total. Commercial FPGAs such as the Zynq devices considered in this work cannot power gate their fabric without losing the device configuration stored in SRAM memory so power gating states are not possible without a full reconfiguration cycle. Taken into account these constraints the next section investigates the energy benefits of the proposed adaptive voltage scaling technology.

B. Energy Scaling

Fig. 9 shows the measurement approach for the energy experiments. The total time T_{total} is fixed and determined by the time needed for the slowest frequency point to obtain one million clock cycles of computation. This value is used as a reference point. As voltage and frequency increase T_{active} reduces since the same amount of clock cycles can be obtained in a smaller amount of time. The time left from subtracting T_{total} and T_{active} is the idle time in which only static power remains. Fig.10 shows the energy analysis and compares it with the nominal energy obtained at a nominal voltage of 1 V. The nominal energy case remains constant for different frequencies since voltage is fixed at 1 V. The maximum performance point is the right most point in Fig. 10 in which the proposed AVS approach doubles the performance for the same amount of energy as the nominal case. The left most point of the figure represents the most energy efficient point in which the AVS points reduces energy by ~65%.

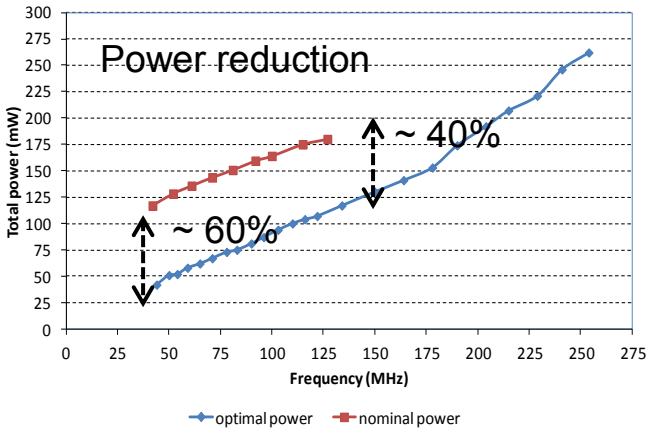

Figure 7. Total power analysis.

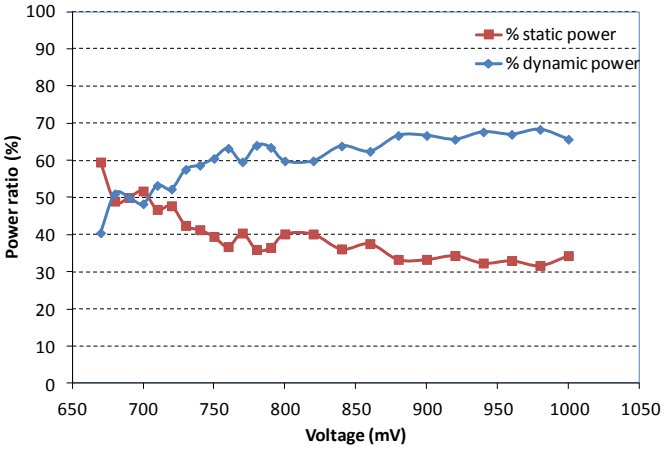

Figure 8. Static and dynamic power ratios in the Zynq 7020 device.

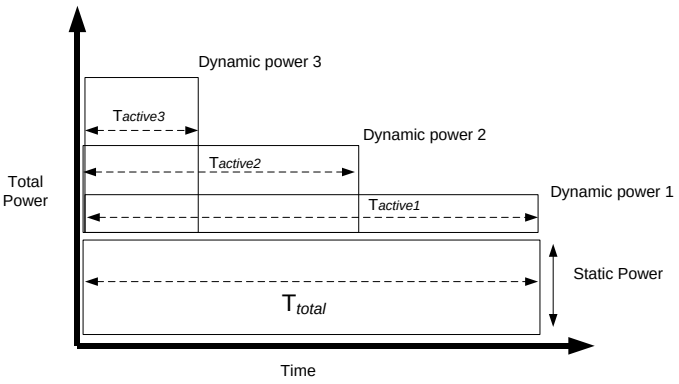

Figure 9. Static and dynamic power timings.

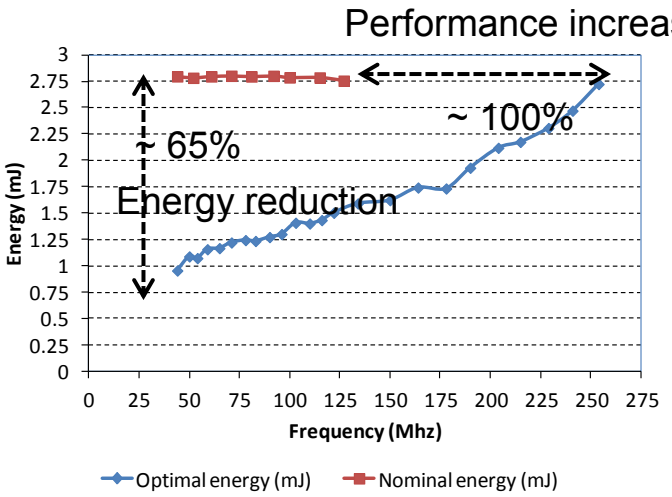

Figure 10. Total energy analysis.

VI. CONCLUSIONS

The considered Zynq devices offer a hybrid computing platform with a hardwire ARM dual-core Cortex A9 processor and a 28 nm FPGA fabric in different voltage domains. This configuration opens the possibility of having software daemons or the OS managing different power and performance configuration points in the FPGA fabric with voltage and frequency scaling. The FPGA fabric maps user-defined accelerators such as the motion estimation processor considered in this work. The adaptive power adaptive architecture is designed to remove timing margins using in-situ timing detectors and includes two main components to control voltage and frequency: the DVS and DFS. The DVS exploits the presence of software programmable voltage regulators via the PMBUS protocol to change voltages at run time while the DFS uses the highly flexible mixed mode clock managers. The availability of the standard PMBUS means that a robust voltage control and monitoring loop can be created with IP blocks without board modifications. The results show that the margins available make these chips a good platform for energy proportional computing. The hybrid device is also interesting because it is possible to map safety critical parts of the application to the hardwired processor while compute intensive accelerators that can tolerate some level of uncertainty could be mapped to the fabric. This uncertainty could originate in the fabric by pushing its performance and energy efficient points validated by the manufacturer. To investigate these uncertainty energy and performance trade-offs is part of our future work.

REFERENCES

1. Kuon, I. and Rose, J. 2007. Measuring the gap between fpgas and asics. Computer-Aided Design of Integrated Circuits and Systems, 26, 2, 203 – 215.
2. Nunez-Yanez, J, Adaptive Voltage Scaling with in-situ Detectors in Commercial FPGAs, Accepted for publication in IEEE transactions on Computers.
3. Rahman, A., Das., Tuan T., and Rahut, A. 2005. Heterogeneous routing architecture for low-power FPGA fabric. In Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005. pp. 183 – 186.
4. Ryan, J. and Calhoun, B. 2010. A sub-threshold fpga with low-swing dual-vdd interconnect in 90nm cmos. (CICC), 2010 IEEE. pp. 1 –4.
5. Li, F., Lin, Y., and He, L. 2004. Vdd programmability to reduce fpga interconnect power. In Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM. pp. 760 – 765.
6. Li, F., Lin, Y., He, L., and Cong, J. 2004. Low-power fpga using pre-defined dual-vdd/dual-vt fabrics. In Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04. ACM, New York, NY, USA, 42–50.
7. Raham A. and Polavarapuv, V. 2004. Evaluation of low-leakage design techniques for field programmable gate arrays. In Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04. ACM, New York, NY, USA, 23–30.
8. Lamoureux, J. and Wilton, S. . On the interaction between power-aware fpga cad algorithms. In Computer Aided Design, 2003. ICCAD-2003. 701 – 708.
9. Lamoureux, J. and Wilton, S. 2007. Clock-aware placement for FPGAs. In Field Programmable Logic and Applications, 2007. FPL 2007. on. 124 –131.
10. Gayasen, A., Tsai, et al. 2004. Reducing leakage energy in fpgas using region constrained placement. In Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays. FPGA '04. ACM, New York, NY, USA, 51–58.
11. Chow, C., Tsui, L., Leong, P., Luk, W., and Wilton, S. 2005. Dynamic voltage scaling for commercial FPGAs. In Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on. 173 –180.
12. Atukem, N. Nunez-Yanez, J... Adaptive Voltage Scaling in a Dynamically Reconfigurable FPGA-Based Platform. ACM Trans. Reconfigurable Technol. Syst. 5, 4, Article 20 (December 2012)
13. Information available at http://www.xilinx.com/support/documentation/application_notes/xapp555-Lowering-Power-Using-VID-Bit.pdf
14. S. Das, et al., Razor II, IEEE J. Solid-State Circuits, pp. 32--48, Jan. 2009.
15. Nunez-Yanez, J.L.; et al "Cogeneration of Fast Motion Estimation Processors and Algorithms for Advanced Video Coding," TVLSI, vol.20, no.3, pp.437-448, 2012